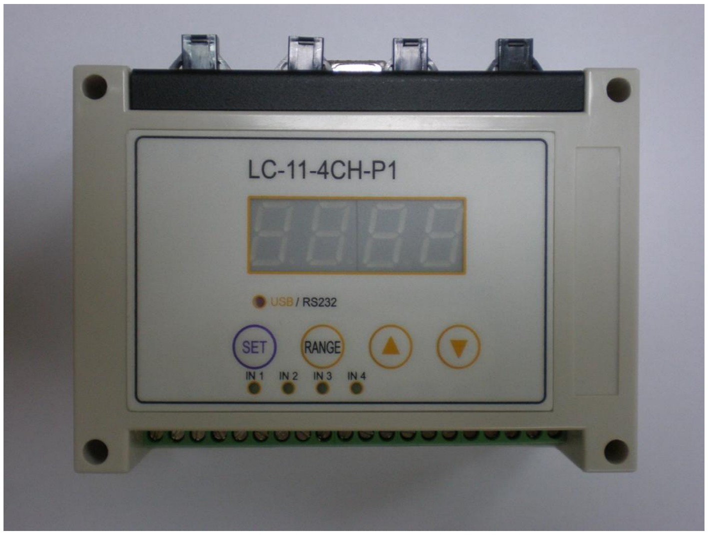
LIGHTING SOLUTION PARTNER



VL-LC-11-4CH-P1
4CH Lighting Controller Unit (Cascade)
(Constant and Trigger Mode)

USER MANUAL

Rev 1.0 May 2011

Contents

Revision Notes

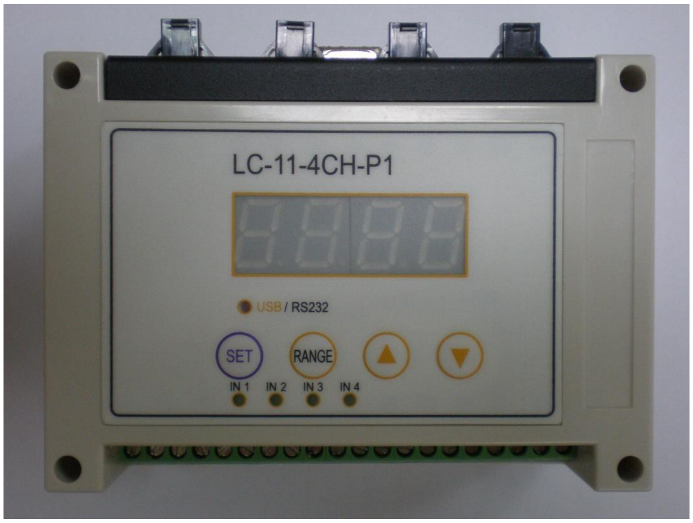
Rev	Date/Author	Comment
1.0	KW/ Jonathan	Cascade - First Release

Hardware

Packing List

Please make sure that the following parts are in the packing list:
- LC-11 lighting controller unit
- USB/RS232 Cable (Optional One)
- Power Supply (Optional)
- 4 LED Lightings (Optional)
- 4 Ext. Cable 1.5 Meter (Optional)

Front Panel



Connections

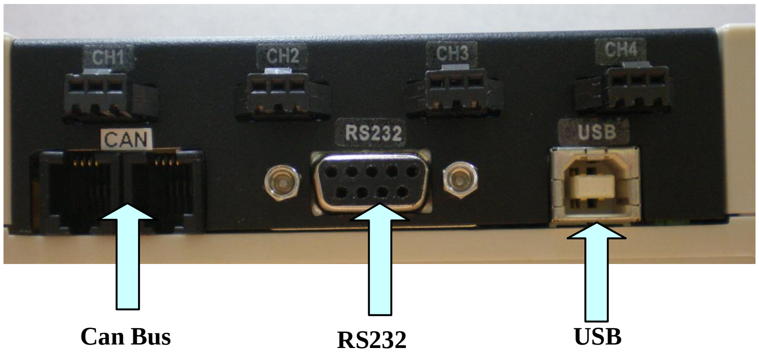
All connections are available on screw terminal blocks. The opto-isolator inputs require a voltage between 5V and 24V DC for a positive logic level. Open circuit or less than 1V gives a negative logic level.

Screw Terminal Block ID	**Function**
VS	Controller power supply +
GND	Controller power supply -
VL	Lighting power supply +
GND	Lighting power supply -
IN1+	Input 1 positive
IN1-	Input 1 negative
IN2+	Input 2 positive
IN2-	Input 2 negative
IN3+	Input 3 positive
IN3-	Input 3 negative
IN4+	Input 4 positive
IN4-	Input 4 negative
CH1+	Channel 1 positive
CH1-	Channel 1 negative
CH2+	Channel 2 positive
CH2-	Channel 2 negative
CH3+	Channel 3 positive
CH3-	Channel 3 negative
CH4+	Channel 4 positive
CH4-	Channel 4 negative

Controller power supply GND and lighting power supply GND are common.

Lighting power supply supplies power to all 4 lighting channels.

USB or RS232 Control:



General Description

Specification

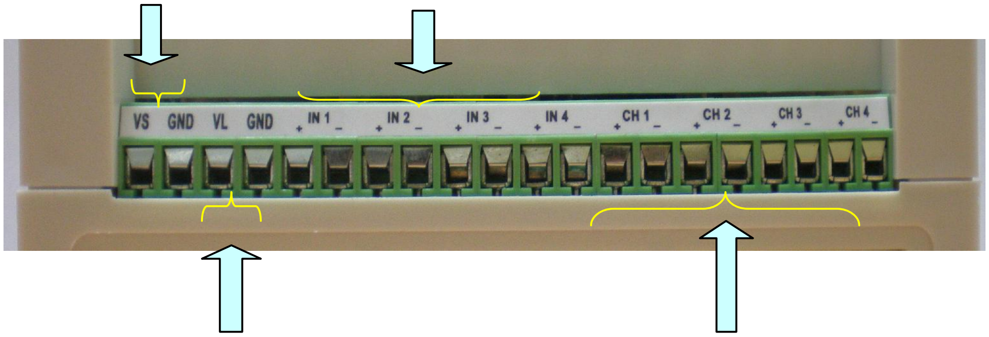
Output	Step	Digital 256 steps (0-255)
	No. of output channel	4
	Voltage	12/24VDC
	Output current	1A per channel
	Light adjustment respond time using button control	Within 420ms, at 24V output
	Input-to-output respond time	Within 50us, at 24V output
Input	Trigger output channel	12V/24V
Display	4-digit 7-segment display	Colour: Red
Power supply	Power rating	7-35VDC
	Current consumption	Approx. 44mA
	Lighting power supply	12/24VDC
Control	Control method	1.Display panel (Buttons) 2. Com Port (RS232) 3. USB

Power Input

The board power connectors should be connected to power supply of voltage 7-35 VDC
The lighting power can be supplied with 12/24V depend on the lighting power rating

Board Input Power
+7-35 VDC

Strobe Input



Lighting Input Voltages

Lighting Output

Controller Mode

1. Constant Mode
2. Trigger Mode
3. Auto Strobe Mode

CONSTANT MODE

- 4 Output Channel (CH1 – CH4)
- The output is continuous current
- Range from 0 – 255

TRIGGER MODE

- 4 Inputs (IN1 – IN4)
- All 4 inputs have a same common of 24V (COM)
- Pull low trigger
- Output is pulsed once per trigger
- One input is used as a trigger (IN1 trigger CH1, IN2 trigger CH2, etc.)
- Pulse Width (0.1 – 999.9ms in 100us steps)
- The output intensity value in trigger mode is the intensity value in constant mode
- To use the trigger mode, set the desired constant value of the channel, then change to trigger mode to set the pulse width.

AUTO STROBE MODE

- Output is pulsed continuously
- Up Time and Down Time (1 – 999ms in 1ms steps)
- The output intensity value in auto strobe mode is the intensity value in constant mode
- To use the auto strobe mode, set the desired constant value of the channel, then change to auto strobe mode to set up time and down time.

The controller has a display panel consisting of a 4-digit seven segment display, four push buttons and an LED indicator.

Operation

The controller LC-11 can be controlled using either the display panel or external control (RS232/USB). Both controls cannot be used at the same time.

The LED labeled “USB/RS232” is an indicator of which control is in use:

1. LED turned on – **USB/RS232 control mode**
2. LED turned off – **Display Panel control mode**

When the power is switched on, the controller is in the control mode last used on the LC-11 controller.

To switch the control, hold down UP and DOWN for about 1 second and observe the change on the LED indicator.

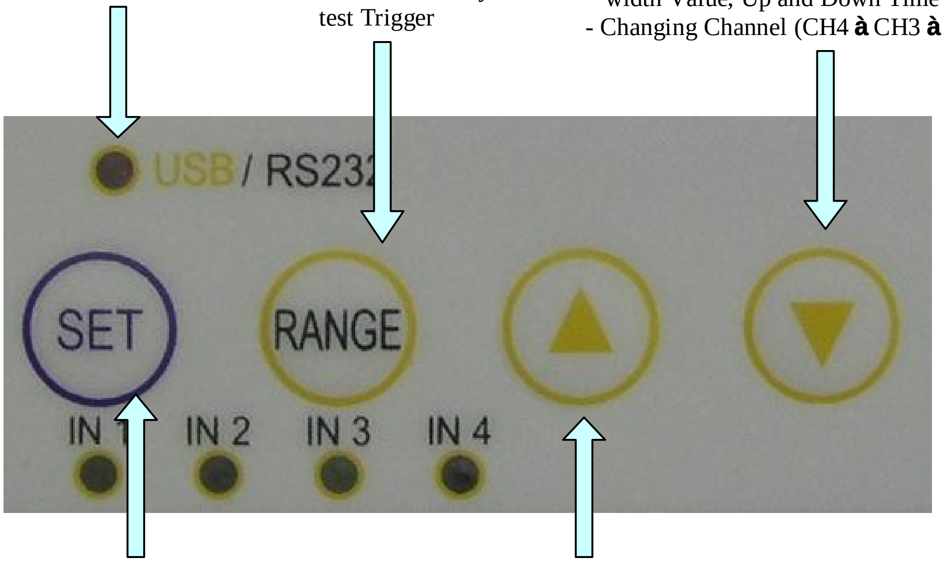
Display Panel Control Mode

In Display Panel Control Mode, user will control the controller by using the 4 Buttons at the Front Panel (Set, Range, Up and Down) for setting Channel, Lighting intensity and trigger pulse width.

USB/RS232
Control Mode LED

Button “RANGE”
is for Moving Digit
for Setting Value
and for manually
test Trigger

Button “DOWN”
- Changing Mode(CON à TRIG à AUTO)
- Minus (-) Lighting Intensity, Trigger pulse width Value, Up and Down Time
- Changing Channel (CH4 à CH3 à CH2 à CH1)

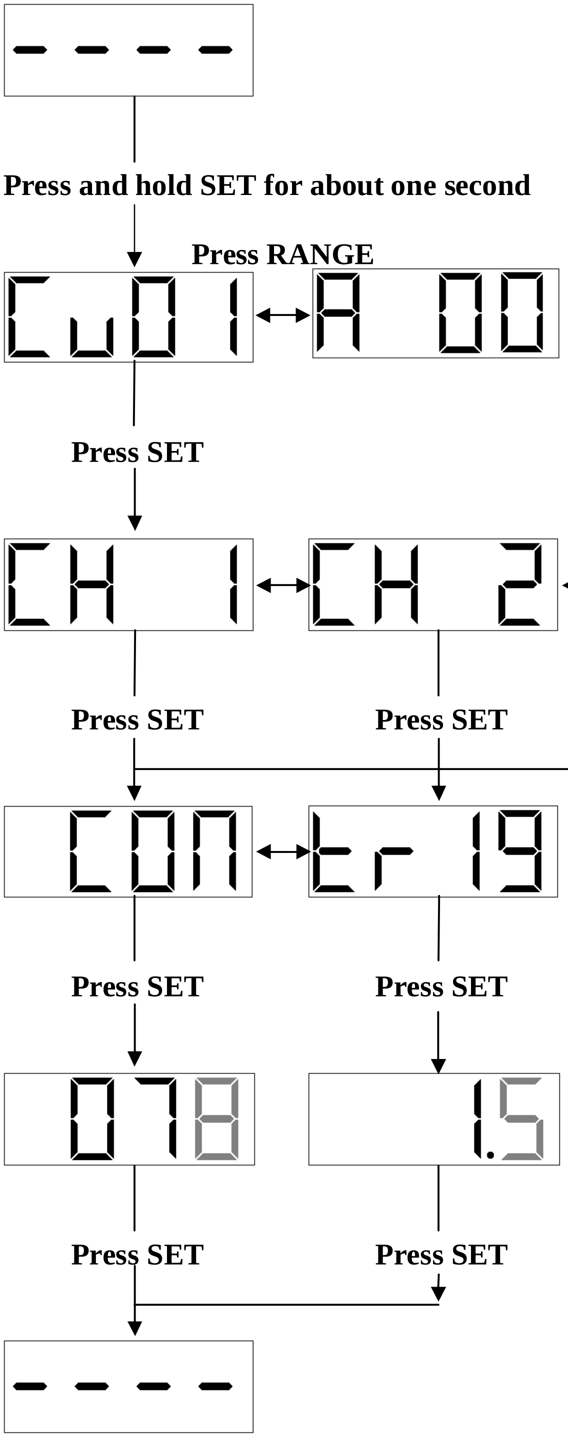


Button “SET” is
for set Parameter
Controller

Button “Up”
- Changing Mode(CON à TRIG à AUTO)
- Add (+) Lighting Intensity, Trigger pulse width Value, Up and Down Time
- Changing Channel (CH1 à CH2 à CH3 à CH4)

Front Panel

Setting up Constant and Trigger Operation using Display Panel



Idle State

Use UP and DOWN to select the current multiplier (1 to 10 which equals to 10 – 100% of max current) for all channels OR change board address for cascade operation

Use UP and DOWN to select which channel to set up (CH1 to CH4)

Use UP and DOWN to select which mode to use (CON or TRIG)

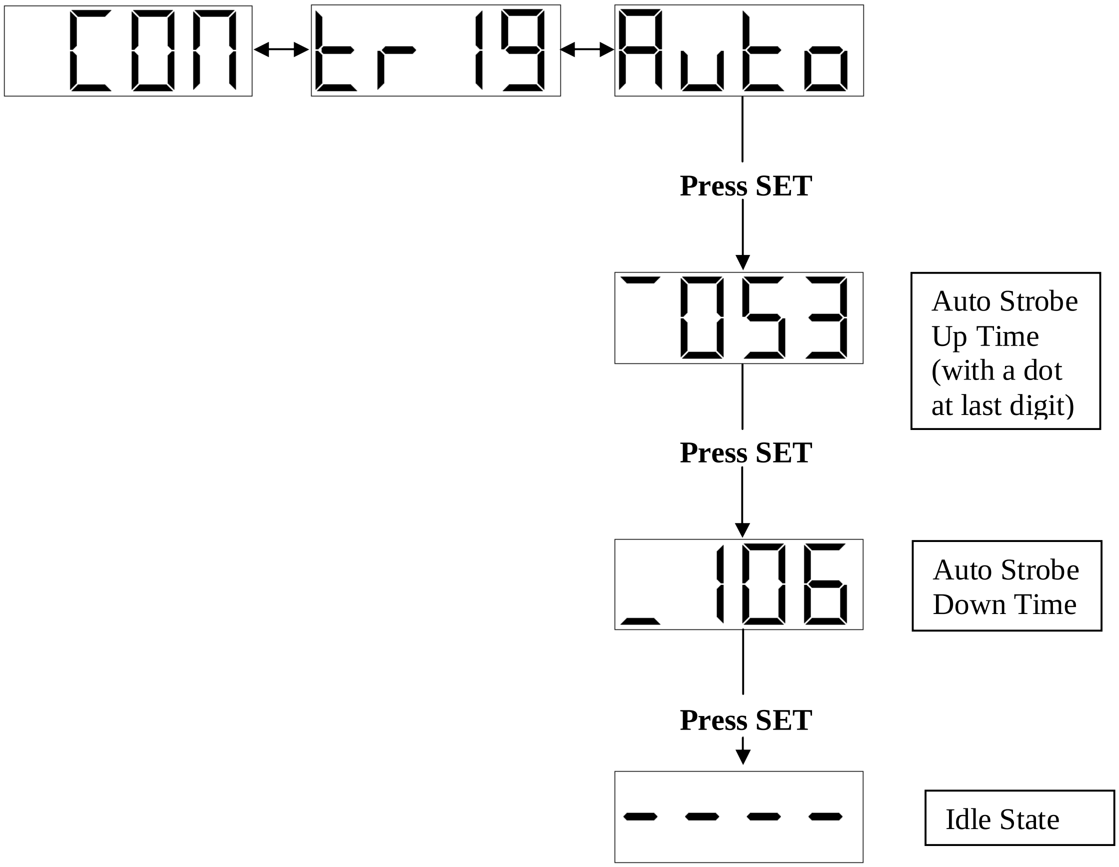
Use UP and DOWN to select the required constant value (0 to 255) and delay in milliseconds (0.0 to 9.9). The delay is set up in multiples of 100us.

Minimum pulse width is 0.1ms, if user set the value as 0, it will automatically set it as 0.1ms.

The “units” digit is flashing. UP or DOWN change the value by one. Press RANGE to select the “tens” and “hundreds” digits.

Back to Idle State

Auto Strobe Mode:

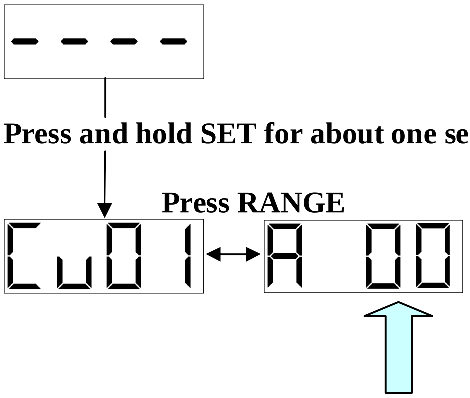


Remark:
Trigger operation can only work when the display is in Idle State.

During Trigger mode, the trigger function of the respective channel can be simulated using the RANGE button. The channel is triggered once when the RANGE button is pressed. This only works when the display is in IDLE mode.

During Auto Strobe mode, the auto strobe function for all channel can be turned ON or OFF using the RANGE button. Only work for channel that is in AUTO mode.

Setting up Board Address for Cascade Operation



Press and hold SET for about one second

Press RANGE

* Use UP and DOWN button to select board address

Board address must be set for controllers used in cascade operation.
Master controller is set as address 00
Slave controller can be set start from address 01

** REMARK: Address should be set to 00 if use as individual controller

Parameters in EEPROM

After the parameters have been set, they are saved into the controller’s EEPROM to retain them after the power to the controller is turned off.

- To reset the parameters

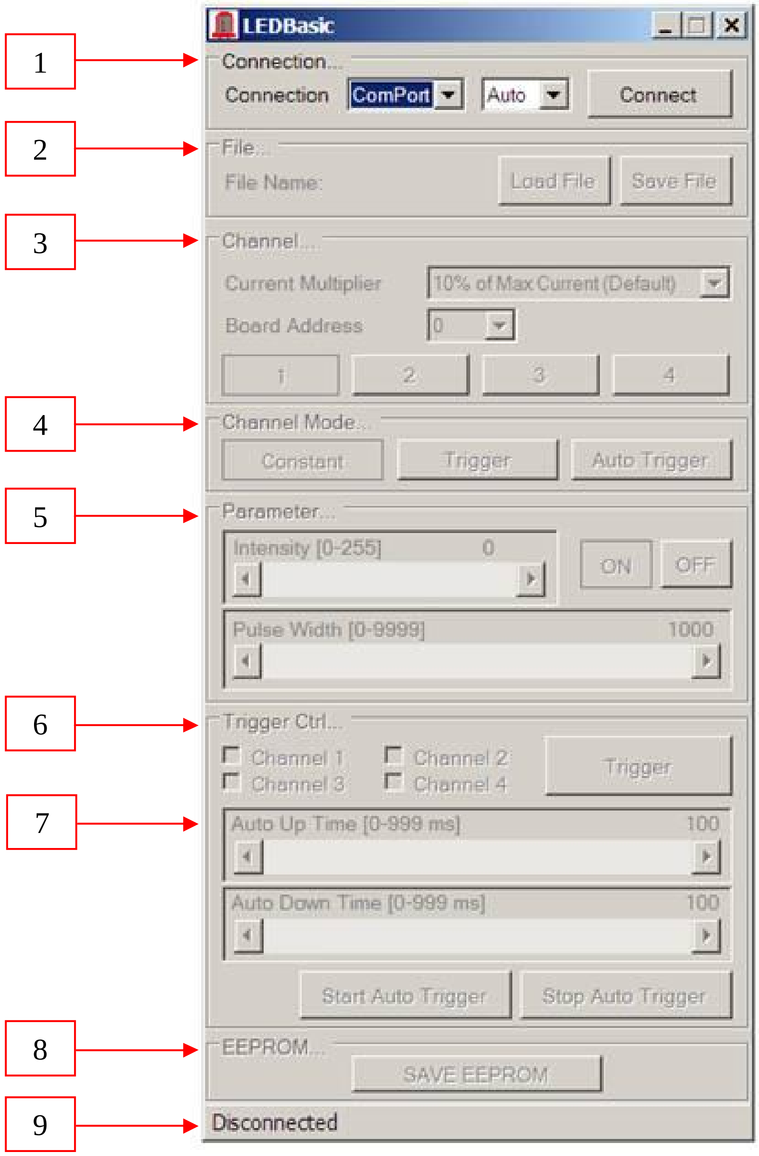
The parameters in the memory can be reset to default values. To do this, first turn off the power of the controller. Hold down the “SET” button while turning on the power, then let go of the button.

USB/RS232 Control Mode

In USB/RS232 Control Mode, the LED on the front panel will turn “On”. Connect the RS232 Cable or USB Cable (does not need USB Driver Installation). The controller can be controlled using the LEDBasic Software.

LEDBasic

1. Copy the folder LEDStudio which includes LEDLib and LEDBasic folder, into drive C:\
3. Run LEDBasic.exe in LEDBasic folder.



Connection ComPort Auto

Current Multiplier 10% of Max Current (Default)

Board Address 0

Intensity [0-255] 0

Pulse Width [0-9999] 1000

Auto Up Time [0-999 ms] 100

Auto Down Time [0-999 ms] 100

Disconnected

LEDBasic GUI

Description

1. **Connection**
   - Select the type of connection to be used - ComPort(RS232) or USB
   - Select the port number where the controller is connected (Auto or 1-256)
   - Connect/Disconnect

2. **File**
   - Load previous configured parameters
   - Save current parameters
   - The configuration is saved as .ini file

3. **Channel**
   - Select current multiplier for all channels (10%, 20%, … , 100% of max current)
   - Select the channel to change the parameter
   - Select board address for cascade operation
   - Channel 1 to 4

4. **Channel Mode**
   - Select mode for channel
   - Constant Mode, Trigger Mode, Auto Strobe Mode

5. **Parameter**
   - Change the light intensity (0-255)
   - Turn light output ON and OFF
   - Change pulse width (0-999.9ms)

6. **Trigger Control**
   - Select the channel to trigger, also use for channel select in Auto Strobe mode
   - Channel 1 to 4
   - Trigger the selected channel(s)

7. **Auto Strobe Control**
   - Change the Auto Strobe Up Time and Down Time (0-999ms)
   - Channel 1 to 4
   - Trigger the selected channel(s)

8. **EEPROM**
   - Load parameters from EEPROM
   - Save parameters into EEPROM

9. **Connection Status**
   - Show the connection status

LC-11 Cascade Operation

The LC-11 Lighting Controller can be used in cascade mode when there are more than 4 lighting output channel to be controlled at the same time.

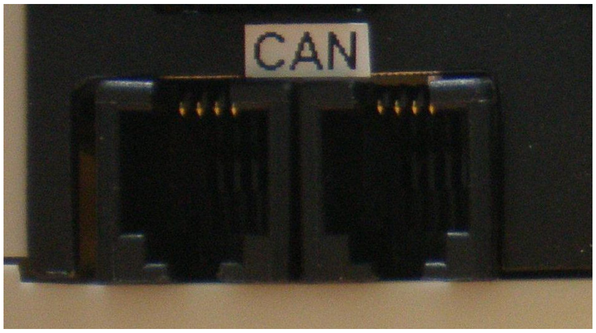
The number of LC-11 controller can be connected in cascade is up to 16 where the address of the controller is pre-set using the button control.

To use LC-11 in cascade,

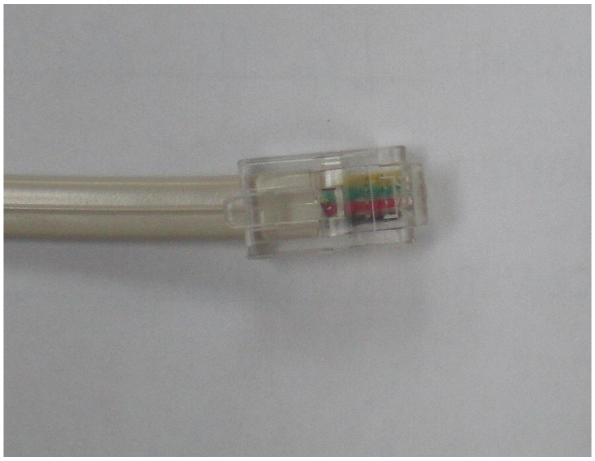
- The address of the master controller must be set to 0 and communicate with the PC using RS232 or USB
- The slave controller address can be set from 1 to 15 and is connected to the master controller using RJ-10

Connection:

- Connect one side of the CAN Bus cable to any of the CAN BUS socket of the main LC-12 controller (address 00) and the other side to CAN BUS Socket of other slave controller (address 01 – 15)
- To control more controllers, just connect them in loop

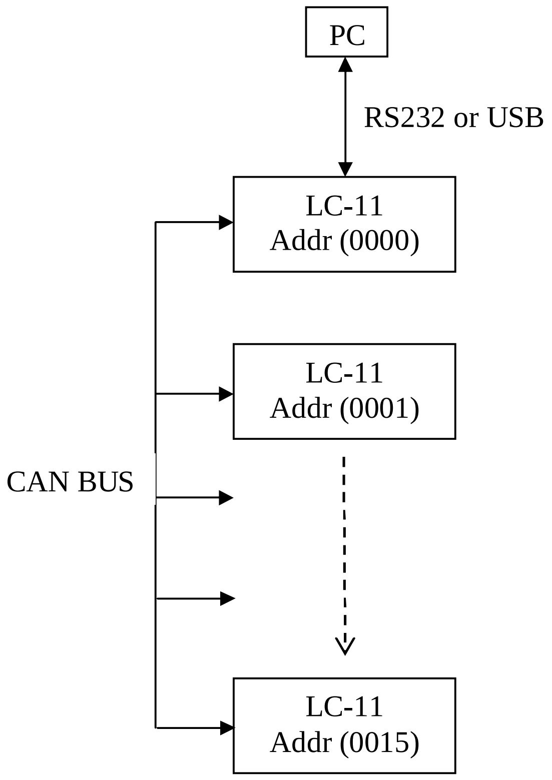


CAN BUS CONNECTOR



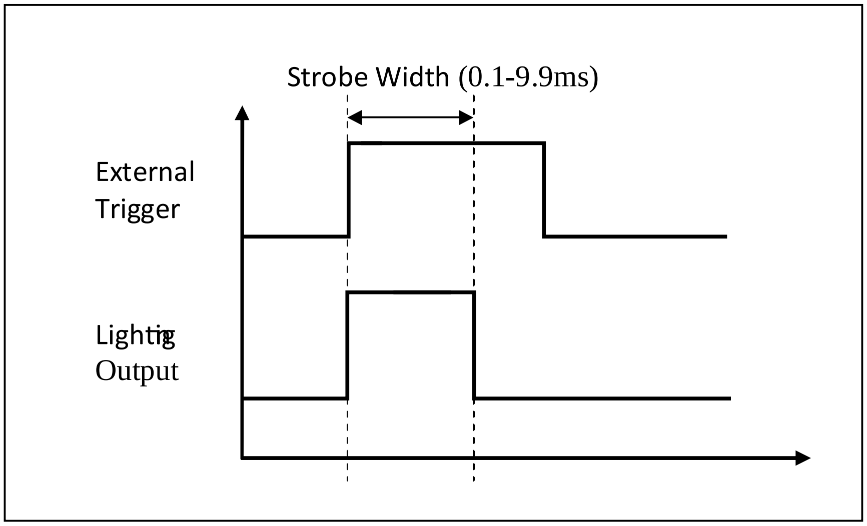
CAN BUS CABLE

CAN BUS multi-drop connection:



Operation Description

Timing Diagram



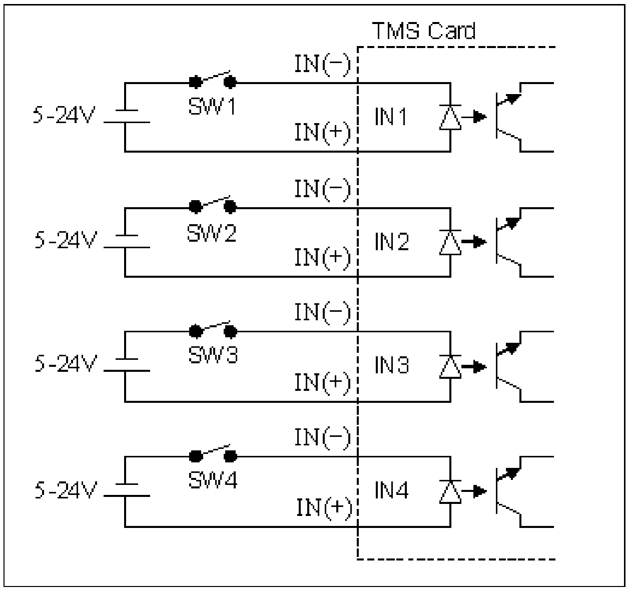
Trigger Timing Diagram

External Input Signal Control

External input signals are optional and can be used ON/OFF the lightings based on the signals (trigger mode). The input signal for a particular channel has no effect if the channel is operating in Constant Mode or when the previous strobe command is still running.

IN1, IN2, IN3 and IN4 are input signals for Channel 1, Channel 2, Channel 3 and Channel 4 respectively. IN(+) is the common positive input and should be connected to 5V – 24V, IN(-) is the common negative input and should be connector to GND as shown in the following figure.

The strobe signal will be acknowledged by the controller on the falling edge or rising edge of the strobe input signal.

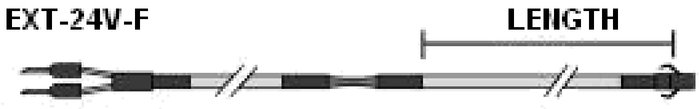


Strobe Input Signals Connection

Constant mode intensity to current table

	Current (mA) (Approximate)									
Intensity	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0	0	0	0	0	0	0	0	0	0	0
10	4	8	12	16	20	24	27	31	35	39
20	8	16	24	31	39	47	55	63	71	78
30	12	24	35	47	59	71	82	94	106	118
40	16	31	47	63	78	94	110	125	141	157
50	20	39	59	78	98	118	137	157	176	196
60	24	47	71	94	118	141	165	188	212	235
70	27	55	82	110	137	165	192	220	247	275
80	31	63	94	125	157	188	220	251	282	314
90	35	71	106	141	176	212	247	282	318	353
100	39	78	118	157	196	235	275	314	353	392
110	43	86	129	173	216	259	302	345	388	431
120	47	94	141	188	235	282	329	376	424	471
130	51	102	153	204	255	306	357	408	459	510
140	55	110	165	220	275	329	384	439	494	549
150	59	118	176	235	294	353	412	471	529	588
160	63	125	188	251	314	376	439	502	565	627
170	67	133	200	267	333	400	467	533	600	667
180	71	141	212	282	353	424	494	565	635	706
190	75	149	224	298	373	447	522	596	671	745
200	78	157	235	314	392	471	549	627	706	784
210	82	165	247	329	412	494	576	659	741	824
220	86	173	259	345	431	518	604	690	776	863
230	90	180	271	361	451	541	631	722	812	902
240	94	188	282	376	471	565	659	753	847	941
250	98	196	294	392	490	588	686	784	882	980
255	100	200	300	400	500	600	700	800	900	1000

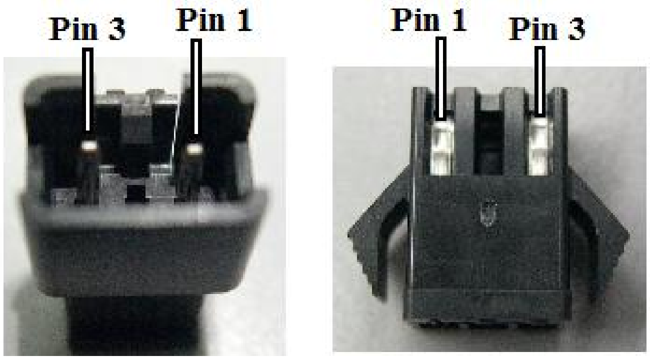
CABLE SELECTION



MODEL	POWER	LENGTH	APPLICATIONS
EXT-24V-F	24V	1M 2M 3M 4M 5M	- Used to connect 24V light to USB / RS232 / KL-4000 / STB / ANG-1CH-P1 / LC-11 Series power supply

LIGHTING CONNECTORS

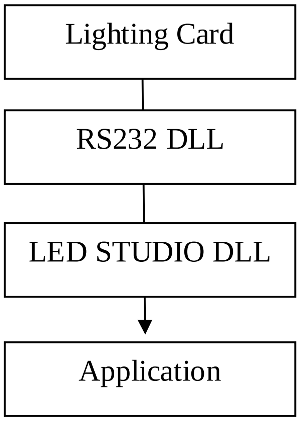
24V-Lighting Connector



24V Connector
Pin 1 —+ve (red wire)
Pin 3 — -ve (white wire)

Pin 1 -- Red wire (+ve)   Pin 3 -- White wire (-ve)

LEDStudio DLL Documentation



DLL Location Path

1. RS232 DLL must be located at C:\LEDStudio\LEDLib\RS232.dll
2. LEDStudio DLL must located at C:\LEDStudio\LEDLib\LEDStudio.dll

Function List

1. function LE_GetVersion(var Value: PChar): Byte; stdcall;
2. function LE_ComportConnect(var ComPort: Byte): Byte; stdcall;
3. function LE_ComportDisConnect(Comport: Byte): Byte; stdcall;
4. function LE_USBConnect(var ProductName: PChar): Byte; stdcall;
5. function LE_USBDisconnect: Byte; stdcall;
6. function LE_SetMultiplier(sMul: Byte): Byte; stdcall;
7. function LE_SetCHMode(sCH, sMode:Byte): Byte; stdcall;
8. function LE_SetConstInt(sInt: Byte): Byte; stdcall;
9. function LE_SetCONSTOnOff(sOn: Byte): Byte; stdcall;
10. function LE_SetStrobeWidth(sWidth: Word): Byte; stdcall;
11. function LE_STROBE(CH1, CH2, CH3, CH4: Boolean): Byte; stdcall;
12. function LE_GETErrMsg(ErrCode: Byte):Pchar ; stdcall;
13. function LE_SetAutoStrobeUpTime(UpTime: Word): Byte; stdcall;
14. function LE_SetAutoStrobeDnTime(DnTime: Word): Byte; stdcall;
15. function LE_AUTOSTROBE(CH1, CH2, CH3, CH4: Boolean): Byte; stdcall;
16. function LE_EEPROM: Byte; stdcall;
17. function LE_SelectAdd(sAdd:Byte): Byte; stdcall;

1. **function LE_GetVersion(var Value: PChar): Byte;**
   Get the version information.
   Parameter:-
   Value: String (by reference)
   - Version number
   Return:-
   Error Code: Byte

2. **function LE_ComportConnect(var ComPort: Byte): Byte;**
   Connect the lighting device via serial com port defined by 'Comport'
   If Comport = 0, LEDStudio will auto detect and connect with Lighting device and will return the comport number connected
   Parameter:-
   ComPort: Byte (by reference)
   - Comport number
   Return:-
   Error Code: Byte

3. **function LE_ComportDisConnect(Comport: Byte): Byte;**
   Disconnect the serial com port with lighting device
   Parameter:-
   Comport: Byte (by value)
   - Comport number
   Return:-
   Error Code: Byte

4. **function LE_USBConnect(var ProductName: PChar): Byte;**
   Connect the lighting device via USB port defined.
   LEDStudio will auto connect USB port with Lighting device and will return product name of the lighting device
   Parameter:-
   ProductName: String (by reference)
   - Product Name
   Return:-
   Error Code: Byte

5. **function LE_USBDisconnect: Byte;**
   Disconnect the lighting device from USB Port
   Parameter:-
   None
   Return:-
   Error Code: Byte

6. **function LE_SetMultiplier(sMul: Byte): Byte;**
   Set the current multiplier of lighting device
   Parameter:-
   sMul: Byte (by value)
   - Current Multiplier
   Range = 1 – 10
   1 = 10% of Max Current (Default)
   2 = 20% of Max Current
   3 = 30% of Max Current
   | |
   9 = 90% of Max Current
   10 = 100% of Max Current
   Return:-
   Error Code: Byte

7. **function LE_SetCHMode(sCH, sMode:Byte): Byte;**
   Set the mode of Channel defined by 'sCH' and 'sMode'.

   Parameter:-
   sCH: Byte (by value)
   - Channel number
   Range = 1 – 4

   sMode: Byte (by value)
   - Channel mode
   Range =
   LEDBasic:
   Mode = 0 => Constant
   Mode = 1 => Trigger
   Mode = 3 => Auto Strobe

   Return:-
   Error Code: Byte

8. **function LE_SetConstInt(sInt: Byte): Byte;**
   Set the Constant Intensity defined by ‘sInt’ for selected channel
   Parameter:-
   sInt: Byte (by value)
   - Constant Intensity
   Range = 0-255
   Return:-
   Error Code: Byte

9. **function LE_SetCONSTOnOff(sOn: Byte): Byte;**
   Set ON or OFF for output of lighting device for selected channel
   Parameter:-
   sOn: Byte (by value)
   - Constant Output ON/OFF
   Range:
   On = 1
   Off = 0
   Return:-
   Error Code: Byte

10. **function LE_SetStrobeWidth(sWidth: Word): Byte;**
    Set the Strobe Pulse Width defined by 'sWidth' for selected channel.
    Parameter:-
    sWidth: Word (by value)
    - Strobe Width
    Range = 0-9999(0-999.9ms)
    Return:-
    Error Code: Byte

11. **function LE_STROBE(CH1, CH2, CH3, CH4: Boolean): Byte;**
    To Trigger the Strobe defined by CH1,CH2,CH3,CH4 with boolean value
    In LEDBasic, use this function to TRIGGER
    Parameter:-
    CH1, CH2, CH3, CH4: Boolean (by value)
    - Channel to Strobe
    Range:
    1 = Trigger
    0 = No trigger
    Return:-
    Error Code: Byte

12. **function LE_GETErrMsg(ErrCode: Byte):Pchar ;**
    Get error message
    Parameter:-
    ErrCode: Byte (by value)
    - Error Code retrieved from called function
    Return:-
    Error Message: String

13. **function LE_SetAutoStrobeUpTime(UpTime: Word): Byte;**
    Set the auto strobe Up Time
    Parameter:-
    UpTime: Word (by value)
    - Autostrobe On Time
    Range = 1-999ms
    Return:-
    Error Code: Byte

14. **function LE_SetAutoStrobeDnTime(DnTime: Word): Byte;**
    Set the auto strobe Down Time
    Parameter:-
    DnTime: Word (by value)
    - Autostrobe Off Time
    Range = 1-999ms
    Return:-
    Error Code: Byte

15. **function LE_AUTOSTROBE(CH1, CH2, CH3, CH4: Boolean): Byte;**
    Turn On/Off the auto strobe
    Parameter:-
    CH1, CH2, CH3, CH4: Boolean (by value)
    - Channel to Autostrobe
    Range =
    1 = On
    0 = Off
    Return:-
    Error Code: Byte

16. **function LE_EEPROM: Byte;**
    To save lighting parameters into the EEPROM
    Parameter:-
    None
    Return:-
    Error Code: Byte

17. **function LE_SelectAdd(sAdd:Byte): Byte; stdcall;**
    Select controller address
    Parameter:-
    sAdd: Byte (by value)
    - Controller address
    Range = 0-15
    Return:-
    Error Code: Byte

Error Code List

Error Code	Desciption	
0	Success No Error Excute	Ensure all integration hardware, cable and lighting device are in proper condition. And also in the correct data range that has been defined. Kindly contact the supplier if the error codes occur.
100	'Err100 - GET_VERSION_ERROR'	
105	'Err105 - COMPORT_CONNECT_ERROR'	
106	'Err106 - NO_COM_DEVICE_DETECTED'	
107	'Err107 - CONNECT_SELECTED_COM_FAIL'	
110	'Err110 - COMPORT_DISCONNECT_ERROR'	
115	'Err115 - USB_CONNECT_ERROR'	
116	'Err116 - NO_USB_DEVICE_DETECTED'	
120	'Err120 - USB_DISCONNECT_ERROR'	
122	‘Err122 - TCPIP_CONNECT_ERROR’	
123	‘Err123 - NO_TCPIP_DEVICE_DETECTED’	
124	‘Err123 - TCPIP_DISCONNECT_ERROR'	
130	'Err130 - SET_CHMODE_ERROR';	
131	'Err131 - INVAILID_CH_NUMBER'	
132	'Err132 - INVAILID_MODE_NUMBER'	
135	'Err135 - SET_MULTIPIER_ERROR'	
140	'Err140 - SET_CONST_INT_ERROR'	
145	'Err145 - SET_CONST_ONOFF_ERROR'	
150	'Err150 - SET_STROBE_INT_ERROR'	
155	'Err155 - SET_STROBE_DELAY_ERROR'	
160	'Err160 - SET_STROBE_WIDTH_ERROR'	
165	'Err165 - SET_STROBE_ODELAY_ERROR'	
170	'Err170 - LE_STROBE_ERROR'	
195	'Err195 - SET_INTENSITY_BANK_ERROR'	
196	'Err196 - BANK_STROBE_ERROR'	
200	'Err200 - SEND_MSG_ERROR'	
201	'Err201 - WRITE_MSG_ERROR'	
202	'Err202 - READ_MSG_ERROR'	
203	'Err203 - READ_MSG_CHECKSUM_ERROR'	
242	'Err242 - COMMAND_ERROR'	lighting device received invalid command
248	'Err248 - DATA_ERROR'	lighting device received invalid data (Data Not In Range)
255	'Err255 - COMMUNICATION_ERROR'	lighting device verify invalid checksum data

EXAMPLE (C#):

DLL IMPORT:

```
const string LEDStudiodll = "C:\\LEDStudio\\LEDLib\\LEDStudio.dll";
[DllImport(LEDStudiodll)]
public static extern byte LE_ComportConnect(ref byte ComPort);
 [DllImport(LEDStudiodll)]
public static extern byte LE_ComportDisConnect(byte Comport);
 [DllImport(LEDStudiodll)]
public static extern byte LE_USBConnect(string ProductName);
 [DllImport(LEDStudiodll)]
public static extern byte LE_USBDisconnect();
 [DllImport(LEDStudiodll)]
public static extern byte LE_SetMultipier(byte sMul);
[DllImport(LEDStudiodll)]
public static extern byte LE_SetCHMode(byte sCH, byte sMode);
[DllImport(LEDStudiodll)]
public static extern byte LE_SetConstInt(byte sInt);
```

CONNECT:

```
byte portnum = 1; // COM1
string productname;
byte err_code;

if (COM_Connect) // COMPORT
{
    if (connected == false)
        err_code = LE_ComportConnect(ref portnum);
    else
        err_code = LE_ComportDisConnect(portnum);
}
else // USB
{
    if (connected == false)
        err_code = LE_USBConnect(productname);
    else
        err_code = LE_USBDisconnect();
}
```

ERROR CHECK:

```
if (err_code != 0)
    MessageBox.Show(LE_GETErrMsg(err_code), "Error");
```

SET MULTIPLIER:

```
byte multiplier = 5; // 50%
err_code = LE_SetMultipier(multiplier);
```

SET CHANNEL & MODE:

```
byte channel = 1; // Channel 1
byte channel_mode = 0; // Constant mode
err_code = LE_SetCHMode(channel, channel_mode);
```

CONSTANT:

```
byte const_int = 100; // Intensity 100
err_code = LE_SetConstInt(const_int);

byte on_off = 1; // On
err_code = LE_SetCONSTOnOff (on_off);
```

STROBE/TRIGGER:

```
UInt16 strobe_width = 1000; // 1ms strobe width
err_code = LE_SetStrobeWidth(strobe_width);

err_code = LE_STROBE(true, false, false, false) // strobe channel 1
```

AUTO STROBE:

```
UInt16 auto_uptime = 250; // 250ms up time
UInt16 auto_dntime = 400; // 400ms down time
err_code = LE_SetAutoStrobeUpTime(auto_uptime)
err_code = LE_SetAutoStrobeDnTime(auto_dntime)

err_code = LE_AUTOSTROBE(true, false, false, false) // turn on channel 1 auto strobe (set
false = turn off)
```

SAVE PARAMTERS IN EEPROM:

```
err_code = LE_EEPROM();
```

EXAMPLE (VC++):

DLL IMPORT:

```
HINSTANCE lib = LoadLibraryA("C:\\LEDStudio\\LEDLib\\LEDStudio.dll");
typedef byte (__stdcall * LE_ComportConnect)(int& ComPort);
LE_ComportConnect ComportConnect =
reinterpret_cast<LE_ComportConnect>(GetProcAddress(lib, "LE_ComportConnect"));
typedef byte (__stdcall * LE_ComportDisConnect)(byte ComPort);
LE_ComportDisConnect ComportDisConnect =
reinterpret_cast<LE_ComportDisConnect>(GetProcAddress(lib,
"LE_ComportDisConnect"));
typedef byte (__stdcall * LE_USBConnect)(char* ProductName);
LE_USBConnect USBConnect = reinterpret_cast<LE_USBConnect>(GetProcAddress(lib,
"LE_USBConnect"));
typedef byte (__stdcall * LE_USBDisconnect)();
LE_USBDisconnect USBDisconnect =
reinterpret_cast<LE_USBDisconnect>(GetProcAddress(lib, "LE_USBDisconnect"));
typedef byte (__stdcall * LE_SetMultiplier)(byte sMul);
LE_SetMultiplier SetMultiplier = reinterpret_cast<LE_SetMultiplier>(GetProcAddress(lib,
"LE_SetMultiplier"));
typedef byte (__stdcall * LE_SetCHMode)(int sCH, int sMode);
LE_SetCHMode SetCHMode = reinterpret_cast<LE_SetCHMode>(GetProcAddress(lib,
"LE_SetCHMode"));
typedef byte (__stdcall * LE_SetConstInt)(int sInt);
LE_SetConstInt SetConstInt = reinterpret_cast<LE_SetConstInt>(GetProcAddress(lib,
"LE_SetConstInt"));
```

CONNECT:

```
int cp = 1; // port number
int& comport = cp; // variable by reference
String ^productname = "";

if (COM_Connect) // COMPORT
{
    if (connected == false)
        err_code = ComportConnect(comport);
    else
        err_code = ComportDisConnect(portnum);
}
else // USB
{
    if (connected == false)
        err_code = USBConnect(productname);
    else
        err_code = USBDisconnect();
}
```

ERROR CHECK:

```
String ^err_msg;
if (err_code != 0) err_msg = GETErrMsg(err_code);
```

SET MULTIPLIER:

```
byte multiplier = 5;
err_code = SetMultiplier(multiplier);
```

SET CHANNEL & MODE:

```
int channel = 1; // CH1
int mode = 0; // constant mode
err_code = SetCHMode(channel, mode);
```

CONSTANT:

```
byte intensity = 100;
err_code = SetConstInt(intensity);
byte on_off = 1; // On
err_code = SetCONSTOnOff (on_off);
```

STROBE/TRIGGER:

```
short strobe_width = 1000; // 1ms strobe width
err_code = SetStrobeWidth(strobe_width);

err_code = STROBE(true, false, false, false); // strobe channel 1
```

AUTO STROBE:

```
short auto_uptime = 250; // 250ms up time
short auto_dntime = 400; // 400ms down time
err_code = SetAutoStrobeUpTime(auto_uptime);
err_code = SetAutoStrobeDnTime(auto_dntime);

err_code = AUTOSTROBE(true, false, false, false); // turn on channel 1 auto strobe (set false = turn off)
```

SAVE PARAMTERS IN EEPROM:

```
err_code = EEPROM();
```

EXAMPLE (VB.NET):

DLL IMPORT:

```
Const LEDStudio As String = "C:\LEDStudio\LEDLib\LEDStudio.dll"
<DllImport(LEDStudio)> _
Public Function LE_ComportConnect(ByRef ComPort As Byte) As Byte
End Function
<DllImport(LEDStudio)> _
Public Function LE_ComportDisConnect(ByVal Comport As Byte) As Byte
End Function
<DllImport(LEDStudio)> _
Public Function LE_USBConnect(ByRef ProductName As String) As Byte
End Function
<DllImport(LEDStudio)> _
Public Function LE_USBDisconnect() As Byte
End Function
<DllImport(LEDStudio)> _
Public Function LE_SetCHMode(ByVal sCH As Byte, ByVal sMode As Byte) As Byte
End Function
<DllImport(LEDStudio)> _
Public Function LE_SetMultipier(ByVal sMul As Byte) As Byte
End Function
<DllImport(LEDStudio)> _
Public Function LE_SetConstInt(ByVal sInt As Byte) As Byte
End Function
```

CONNECT:

```
Dim comport As Byte = 1 ‘COM1
Dim productname As String = ""

If (COM_Connect) Then ‘COMPORT
    If (connected == false)
        err_code = LE_ComportConnect (comport)
    Else
        err_code = LE_ComportDisConnect (comport)
    End If
Else ‘USB
    If (connected == false)
        err_code = LE_USBConnect (productname)
    Else
        err_code = LE_USBDisconnect ()
    End If
End If
```

ERROR CHECK:

```
Dim err_msg As String
If Not err_code = 0 Then
    err_msg = LE_GETErrMsg(err_code)
```

SET MULTIPLIER:

```
Dim multiplier As Byte = 5 ‘50% of max current
err_code = LE_SetMultiplier(multiplier)
```

SET CHANNEL & MODE:

```
Dim channel As Integer = 1 ‘CH1
Dim mode As Integer = 0 ‘constant mode
err_code = LE_SetCHMode(channel, mode)
```

CONSTANT:

```
Dim intensity as Byte = 100
err_code = LE_SetConstInt(intensity)

Dim on_off As Byte = 1 ‘On
err_code = LE_SetCONSTOnOff (on_off)
```

STROBE/TRIGGER:

```
Dim strobe_width as Short = 1000 ‘1ms strobe width
err_code = LE_SetStrobeWidth(strobe_width)
err_code = LE_STROBE(true, false, false, false) ‘strobe channel 1
```

AUTO STROBE:

```
Dim auto_uptime As Short = 250 ‘250ms up time
Dim auto_dntime As Short = 400 ‘400ms down time
err_code = LE_SetAutoStrobeUpTime(auto_uptime)
err_code = LE_SetAutoStrobeDnTime(auto_dntime)
err_code = LE_AUTOSTROBE(true, false, false, false) ‘turn on channel 1 auto strobe (set
false = turn off)
```

SAVE PARAMTERS IN EEPROM:

```
err_code = LE_EEPROM()
```